

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-204639

(43)Date of publication of application : 13.08.1993

(51)Int.Cl.

G06F 9/38

(21)Application number : 04-014578

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 30.01.1992

(72)Inventor : INOUE MASAO

## (54) INFORMATION PROCESSOR

### (57)Abstract:

PURPOSE: To shorten the execution time of an instruction by including a field indicating information relating to instruction dependent relation in an instruction buffer.

CONSTITUTION: Three instructions mul, mul, and add in addresses 0x04 to 0x0c are registered in entries #1 to #3 in respective entry states of the 3rd cycle instruction buffer in a timing chart. The add instruction rsl of the entry #3 indicates the existence of data dependent relation between the preceding instruction and the impossibility of execution start in the succeeding cycle by setting up the 1st bit flag in a dep field. The mul instruction of the entry #2 indicates the existence of data dependent relation between the succeeding instruction by a dep-inst field. When one instruction has data dependent relation between the succeeding instruction, execution is started from the instruction having the dependent relation even when the address of the instruction concerned is larger than that of the other instruction.

colid		dp	rs1	rs2	rcall:	per	hap12	ok	mi	mod	drp
0	0	1									inst
1	1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	0
2	1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	0
3	1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	rs1	0

rejection]

[Kind of final disposal of application other than  
the examiner's decision of rejection or  
application converted registration]

[Date of final disposal for application]

[Patent number] 2906799

[Date of registration] 02.04.1999

[Number of appeal against examiner's  
decision of rejection]

[Da

(19)日本国特許庁(JP)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-204639

(43)公開日 平成5年(1993)8月13日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 9/38

識別記号

3 1 0 F 9290-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数2(全 8 頁)

(21)出願番号 特願平4-14578

(22)出願日 平成4年(1992)1月30日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 井上 雅夫

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(74)代理人 弁理士 小鍛治 明 (外2名)

(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】 同一の演算装置に対して複数の命令が演算の実行を開始できる状態にあった場合、プログラム全体の命令実行時間が短くなるように命令を選択することである。

【構成】 命令バッファ30に命令間のデータ依存に関する情報を示すフィールドを設けることにより、同一の演算装置に対して複数の命令が演算の実行を開始できる状態にあった場合、データ依存関係の存在する命令から先に実行を開始する。このように、命令バッファに命令の依存関係に関する情報を示すフィールドを設けることにより、命令の実行時間を短縮できる。

	valid	op	rs1	rs2	result_ptr	dep[2]	ok	mul	add	dep_inat
#0	0									
#1	1	mul	r3	r4	#1	0 0	1	1	0	0
#2	1	mul	r11	r12	#2	0 0	1	1	0	1
#3	1	add	r13	r14	#3	1 0	0	0	1	0

## 1

## 【特許請求の範囲】

【請求項1】複数の命令を供給する命令記憶装置と、前記命令記憶装置から供給された複数の命令を解読する命令解読装置と、前記命令解読装置で解読された複数の命令を一時的に蓄える命令バッファと、前記命令バッファから供給された命令を実行する複数の演算装置と、前記複数の演算装置で実行された命令の実行結果を一時的に蓄える結果バッファと、命令のオペランドを読み出した命令の実行結果を書き込むためのレジスタファイルと、供給された命令間のデータ依存関係を調べる依存検出装置とを備えた情報処理装置において、前記命令バッファに蓄えられた命令の中から次のサイクルで実行する命令を選択する際、実行可能な命令が複数あり前記複数の実行可能な命令が同一の演算装置を使用する場合、前記複数の実行可能な命令のうちデータ依存関係の存在する命令を先に実行する手段を備えたことを特徴とする情報処理装置。

【請求項2】請求項1記載の複数の実行可能な命令のうちデータ依存関係の存在する命令を先に実行する手段として、命令バッファの各エントリに前記命令バッファ内の他の命令との間でデータ依存関係が存在することを示すフィールドを設けたことを特徴とする情報処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は情報処理装置に関し、具体的には並列に命令を実行する情報処理装置における命令の実行順序の制御に関するものである。

## 【0002】

【従来の技術】従来の情報処理装置は、逐次的に命令をアドレスの順番に従って実行する。情報処理装置のデータ処理能力を増大させるために、複数の演算装置を備え、複数の演算装置に並列に複数の命令を分配して、アドレスの順番によらず実行可能なものから命令の実行を行なう（以降、これをout-of-order実行と呼ぶ）並列処理が実施されている。

【0003】図5は従来の技術を用いたout-of-order実行を行なう情報処理装置のブロック図である。10は命令を記憶する命令記憶装置であり、1サイクルに2つの命令をその命令のアドレスと共に命令バス11、12を通じて供給する。20は命令を解読する命令解読装置であり、1サイクルに2つの命令の解読を行ない、命令バッファ30や結果バッファ60に制御信号バス21、22を通じて命令を実行するための制御信号を供給する。命令バッファ30では命令解読装置から供給された制御信号を一旦蓄え、適当なタイミングで演算装置に制御信号バス31、32を通じて制御信号を供給する。この制御信号は、命令の種別を表す信号とオペランドレジスタアドレスを含む。40は乗除算命令を実行する乗除算装置であり、命令バッファ30からの制御信

## 2

号により、レジスタファイル70あるいは結果バッファ60から供給されたオペランドデータ71、72を用いて乗除算演算を行ない、演算結果41を出力する。また演算した結果をすぐ他の演算器が使う場合には、演算結果41はセクタ42を通じてオペランドバス71、72、73、74にバイパスされる。50は加減算命令を実行する加減算装置であり、命令バッファ30からの制御信号により、レジスタファイル70あるいは結果バッファ60から供給されたオペランドデータ73、74を用いて加減算演算を行ない、演算結果51を出力する。また演算した結果をすぐ他の演算器が使う場合には、演算結果51はセクタ52を通じてオペランドバス71、72、73、74にバイパスされる。

【0004】60は乗除算装置40および加減算装置50で実行された命令の演算結果を一時的に蓄えておく結果バッファである。命令が情報処理装置に供給されたら命令解読装置20で解読されたあと、その命令のアドレスを示すプログラムカウンタとディネーションレジスタアドレス番号は、命令バッファ30に命令の種別を表す信号とオペランドレジスタアドレスをストアされると同じタイミングで結果バッファ60にアドレスの順番にストアされる。結果バッファ60では、複数の演算装置で並列に演算が行なわれているとき大きい方のアドレスの命令が小さい方のアドレスの命令より早く演算が完了した場合には、小さい方のアドレスの演算の完了を待ち、適当なタイミングでレジスタファイル70に演算結果をアドレスの順番に演算結果バス61、62を通じてストアする。このことにより、out-of-order実行を行なう情報処理装置においても正しいプログラムの実行結果が常に得られる。このとき制御信号バス21、22を通じて供給され蓄えられていたデスティネーションレジスタアドレス番号も演算結果と共にレジスタファイル70に供給され、どのレジスタに演算結果をストアするのかを指定する。また必要に応じて演算装置40、50にレジスタファイルの代わりにオペランドデータをオペランドバス71、72、73、74を通じて供給する。

【0005】70はオペランドデータを供給したり、演算結果を最終的に蓄えておくためのレジスタファイルである。80は、命令間のデータ依存関係を調べるための依存検出装置であり、供給された命令のオペランドレジスタアドレスと供給された命令のデスティネーションレジスタアドレスあるいは結果バッファ60に蓄えられているデスティネーションレジスタアドレスとを比較して、データ依存がある場合には命令バッファ30のデータ依存関係を示すフィールドにフラグを立て、結果バッファからの依存解消信号64を受け取るまで命令の実行開始を待たせる。

【0006】図3は図5に示す従来の技術における情報処理装置の構成要素である命令バッファの構成図であ

## 3

り、各エントリーが1命令分の情報を蓄える。図3において、`valid`はそのエントリーが有効であることを示す。`op`は命令種別を示す。`rs1`、`rs2`はそれぞれ第一および第二のオペランドのレジスタ番号を示す。`result_ptr`はその命令の結果バッファ60における対応するエントリーの番号を示し、その命令が実行を開始したら演算器に渡され、演算結果を結果バッファ60にストアするときに使われる。`dep`はその命令のオペランドレジスタ`rs1`、`rs2`にデータ依存関係があるかどうかを示す2ビットのフラグである。`ok`はそのエントリーの命令が次のサイクルにおいて実行可能かどうかを示す。`mul`、`add`はそれぞれそのエントリーの命令が乗除算命令かあるいは加減算命令かどうかを示す。

【0007】図6は図5に示す従来の技術における情報処理装置の構成要素である結果バッファの説明図であり、各エントリーが1命令分の演算結果情報を蓄える。図6において、`PC`はその命令のアドレス情報を示し、ある命令で障害が発生したとき、`OS`の障害回復作業が完了したあと、障害の発生した命令の次の命令から命令の実行を再開したいときに使用する。`valid`はそのエントリーが有効であることを示す。`rd`はデスティネーションレジスタ番号を示す。`d0~d3`は、命令間のデータ依存関係を示す。`d0`は、命令バッファの#0エントリーの命令が結果バッファに登録されている命令のうち、どのエントリーの命令の演算結果をどのオペランドが必要としているかを示す2ビットのフラグである。`d1`、`d2`、`d3`も同様に、それぞれ命令バッファの#1、#2、#3エントリーの命令と結果バッファに登録されている命令との間のデータ依存関係を示す。`results`はそのエントリーの命令の実行結果を示す。`dv`は演算結果が`results`フィールドに存在するかどうかを示す。演算装置から送られてくる演算結果は、該当するエントリーの命令の`results`フィールドにストアされると同時に、そのエントリーの命令にデータ依存関係が存在するかどうか調べ、もしデータ依存関係が存在していて`d0~d3`のいずれかのエントリーにフラグが立っている場合には、そのエントリーの命令の実行結果を待っている命令が命令バッファ30に存在しているので、依存解消信号64により命令バッファ30の該当するエントリーの`dep`フィールドのフラグの立っているビットをクリアする。

【0008】図4は図5に示す従来の技術における情報処理装置の動作を示すタイミングチャート図である。図4において、`D`は命令解読装置20において命令の解読を行なうステージ、`B`は命令バッファ30において演算の実行開始を待つ状態を示すステージ、`L`は演算のオペランドデータをレジスタファイル70あるいは結果バッファ60からオペランドバス71~74を通じて読み出すステージ、`E`は乗除算装置40あるいは加減算装置50

## 4

0において演算を実行するステージ、`S`は結果バッファ60に命令の演算結果を書き込むステージ、`R`はレジスタファイル70に命令の演算結果を書き込むステージを示す。図5に示す従来の技術における情報処理装置においては、乗除算命令、加減算命令ともにEステージに2サイクルを要する。

【0009】図4では、4つの命令から成る命令シーケンスがアドレスと共に示されている。アドレス0x00の乗算命令`mul`の演算結果をアドレス0x04の乗算命令`mul`がオペランドとして用いるデータ依存関係が存在する。したがって、アドレス0x00の乗算命令`mul`の演算結果41は、サイクル4においてアドレス0x04の乗算命令のオペランドとして、セクタ42、オペランドバス71を通じて乗除算装置40にバイパスされる。またアドレス0x08の乗算命令`mul`の演算結果をアドレス0x0cの加算命令`add`がオペランドとして用いるデータ依存関係が存在する。したがって、アドレス0x08の乗算命令`mul`の演算結果41は、サイクル8においてアドレス0x0cの加算命令のオペランドとして、セクタ42、オペランドバス73を通じて加減算装置50にバイパスされる。

【0010】以上のように構成された従来の技術による情報処理装置について、以下図3、図4、図5及び図6を用いて次のサイクルにおいて実行する命令を決めるときの動作を説明する。

【0011】図3は図4のタイミングチャートにおける第3サイクル目における命令バッファの各エントリーの状態を示すものであって、アドレス0x04~0x0cの`mul`、`mul`、`add`の3つの命令がエントリー#1、#2、#3に登録されている。この3つの命令の状態は、図4のタイミング図におけるBステージに相当し、命令の実行開始を待っている。またエントリー#3の`add`命令の第一のオペランド`rs1`は、前の命令との間でデータ依存関係があり、次のサイクルでは実行開始できないことが`dep`フィールドの第一ビットのフラグが立っていることで示されている。

【0012】図6は図4のタイミングチャートにおける第3サイクル目における結果バッファの各エントリーの状態を示すものであって、アドレス0x00~0x0cの`mul`、`mul`、`mul`、`add`の4つの命令がエントリー#0、#1、#2、#3に登録されている。この4つの命令はいずれも演算実行を完了していないので、`dv`フィールドは全て0になっている。

【0013】図3においてつぎのサイクルで実行可能な命令はエントリー#1の`mul`命令とエントリー#2の`mul`命令であることが、`ok`フィールドにフラグが立っていることにより示されている。また`mul`フィールドにフラグが立っていることにより示されているように、エントリー#1とエントリー#2の2つの命令は共に同一の演算装置（乗除算装置40）を使用する。

## 5

【0014】従来の情報処理装置においては、同一の演算装置を使用する実行可能な命令が複数個ある場合には、アドレスの小さい方の命令から実行を開始させるので、この場合、エントリー#1のmul命令がサイクル4でLステージから実行を開始する。この結果図4に示すように、4つの命令の演算実行のためのサイクル数は、3サイクル目から10サイクル目までの8サイクルを要する。

## 【0015】

【発明が解決しようとする課題】しかしながら上記のような構成では、同一の演算装置に対して複数の命令が演算の実行を開始できる状態にあった場合、アドレスの小さい方の命令から順番に実行されるため、アドレスの大きい方の命令を先に実行したほうが全体の命令の実行サイクル数が少なくなる場合でもアドレスの小さい方の命令から実行を開始しなければならないという問題点を有していた。

【0016】本発明は上記問題点に鑑み、同一の演算装置に対して複数の命令が演算の実行を開始できる状態にあった場合、全体の命令実行時間が短くなるように、つぎのサイクルで実行を開始する命令を決める機構を提供するものである。

## 【0017】

【課題を解決するための手段】上記問題点を解決するために本発明の情報処理装置は、複数の命令を供給する命令記憶装置と、前記命令記憶装置から供給された複数の命令を解釈する命令解釈装置と、前記命令解釈装置で解釈された複数の命令を一時的に蓄える命令バッファと、前記命令バッファから供給された命令を実行する複数の演算装置と、前記複数の演算装置で実行された命令の実行結果を一時的に蓄える結果バッファと、命令のオペランドを読み出したり命令の実行結果を書き込むためのレジスタファイルと、供給された命令間のデータ依存関係を調べる依存検出装置とを備えた情報処理装置において、前記命令バッファに蓄えられた命令の中から次のサイクルで実行する命令を選択する際、実行可能な命令が複数あり前記複数の実行可能な命令が同一の演算装置を使用する場合、前記複数の実行可能な命令のうちデータ依存関係の存在する命令を先に実行する手段を備えたものである。

## 【0018】

【作用】本発明は上記した構成によって、同一の演算装置に対して複数の命令が演算の実行を開始できる状態にあった場合、命令バッファ内の命令の依存関係を示すフィールドの内容を調べ、データ依存関係の存在する命令を先に実行する。

## 【0019】

【実施例】以下本発明の一実施例の情報処理装置について、図面を参照しながら説明する。

【0020】図1は本発明の実施例における命令バッ

## 6

ファであり、図5に示す従来の技術による情報処理装置において、図3に示すその構成要素である命令バッファを改良したものである。図1において、validはそのエントリーが有効であることを示す。opは命令種別を示す。rs1、rs2はそれぞれ第一および第二のオペランドのレジスタ番号を示す。result\_ptrはその命令の結果バッファ60における対応するエントリーの番号を示し、その命令が実行を開始したら演算器に渡され、演算結果を結果バッファ60にストアするときに使われる。depはその命令のオペランドレジスタrs1、rs2にデータ依存関係があるかどうかを示す2ビットのフラグである。okはそのエントリーの命令が次のサイクルにおいて実行可能かどうかを示す。mul、addはそれぞれそのエントリーの命令が乗除算命令かあるいは加減算命令かどうかを示す。

【0021】dep\_instは、そのエントリーの命令が命令バッファ内の他の命令とデータ依存関係にあり、そのエントリーの命令の実行結果を他の命令が待っている状態にあることを示す。

【0022】図2は本発明の実施例における情報処理装置の動作を示すタイミングチャート図である。図2において、Dは命令解釈装置20において命令の解釈を行なうステージ、Bは命令バッファ30において演算の実行開始を待つ状態を示すステージ、Lは演算のオペランドデータをレジスタファイル70あるいは結果バッファ60からオペランドバス71~74を通じて読み出すステージ、Eは乗除算装置40あるいは加減算装置50において演算を実行するステージ、Sは結果バッファ60に命令の演算結果を書き込むステージ、Rはレジスタファイル70に命令の演算結果を書き込むステージを示す。本発明の実施例における情報処理装置においては、乗除算命令、加減算命令ともにEステージに2サイクルを要する。

【0023】図2では、4つの命令から成る命令シーケンスがアドレスと共に示されている。アドレス0x00の乗算命令mulの演算結果をアドレス0x04の乗算命令mulがオペランドとして用いるデータ依存関係が存在する。したがって、アドレス0x00の乗算命令mulの演算結果41は、サイクル6においてアドレス0x04の乗算命令のオペランドとして、結果バッファからオペランドバス71を通じて乗除算装置40にパイパスされる。またアドレス0x08の乗算命令mulの演算結果をアドレス0x0cの加算命令addがオペランドとして用いるデータ依存関係が存在する。したがって、アドレス0x08の乗算命令mulの演算結果41は、サイクル6においてアドレス0x0cの加算命令のオペランドとして、セクタ42、オペランドバス73を通じて加減算装置50にパイパスされる。

【0024】以上のように構成された従来の技術による情報処理装置について、以下図1及び図2を用いて次の

7

サイクルにおいて実行する命令を決めるときの動作を説明する。図1は図2のタイミングチャートにおける第3サイクル目における命令バッファの各エントリーの状態を示すものであって、アドレス0x04~0x0cのmul、mul、addの3つの命令がエントリー#1、#2、#3に登録されている。この3つの命令の状態は、図3のタイミング図におけるBステージに相当し、命令の実行開始を待っている。またエントリー#3のadd命令のrs1は、前の命令との間でデータ依存関係があり、次のサイクルでは実行開始できないことがdepフィールドの第一ビットのフラグが立っていることで示されている。またエントリー#2のmul命令は後に続く命令との間でデータ依存関係にあることがdep\_instフィールドにより示されている。

【0025】図2においてつぎのサイクルで実行可能な命令はエントリー#1のmul命令とエントリー#2のmul命令であることが、okフィールドにフラグが立っていることにより示されている。またmulフィールドにフラグが立っていることにより、エントリー#1とエントリー#2の2つの命令は共に同一の演算装置（乗除算装置40）を使用する。

【0026】本発明における情報処理装置においては、同一の演算装置を使用し、次のサイクルにおいて実行可能な命令が複数個ある場合において、一方の命令が後に続く命令との間でデータ依存関係にある場合にはその命令のアドレスが他方の命令のアドレスより大きい場合でも依存関係のある命令から実行を開始させるので、この場合エントリー#2のmul命令がサイクル4でLステージから実行を開始する。この結果、図2に示す通り、4つの命令の演算実行のためのサイクル数は、3サイクル目から8サイクル目までの6サイクルを要し、図4に示す従来の技術による情報処理装置に比べて演算実行サイクル数が2サイクル短い。以上のように本発明の実施例によれば、命令バッファ内に命令の依存関係に関する情報を示すフィールドを設けることにより、プログラム全体の命令の実行時間を短くすることができる。

【0027】なお、本発明の実施例において、並列に実行できる命令の数は2命令としたが、1サイクルに1命

8

令しか実行を開始できない情報処理装置においても同様の機構を設けて命令の実行時間を短縮できる。また3命令以上並列に実行開始できる情報処理装置においても同様の機構を設けることができることは言うまでもない。

【0028】

【発明の効果】以上のように本発明によれば、命令バッファに命令の依存関係に関する情報を示すフィールドを設けることにより、命令の実行時間を短縮できる。

【図面の簡単な説明】

10 【図1】本発明の実施例における情報処理装置の命令バッファ構成図

【図2】同実施例における情報処理装置のタイミング図

【図3】従来の情報処理装置の命令バッファ構成図

【図4】従来の情報処理装置のタイミング図

【図5】従来の情報処理装置のブロック図

【図6】従来の情報処理装置の結果バッファ構成図

【符号の説明】

- 10 命令記憶装置
- 11、12 命令バス
- 20 命令解読装置
- 21、22 制御信号バス
- 30 命令バッファ
- 31、32 制御信号バス
- 40 乗除算装置
- 41 演算結果バス
- 42 セレクタ
- 50 加減算装置
- 51 演算結果バス
- 52 セレクタ
- 30 60 結果バッファ
- 61、62 演算結果バス
- 63 デスティネーションレジスタアドレス番号
- 64 データ依存解消信号
- 70 レジスタファイル
- 71、72、73、74 オペランドバス
- 80 依存検出装置
- 81、82 データ依存検出信号

【图 4】

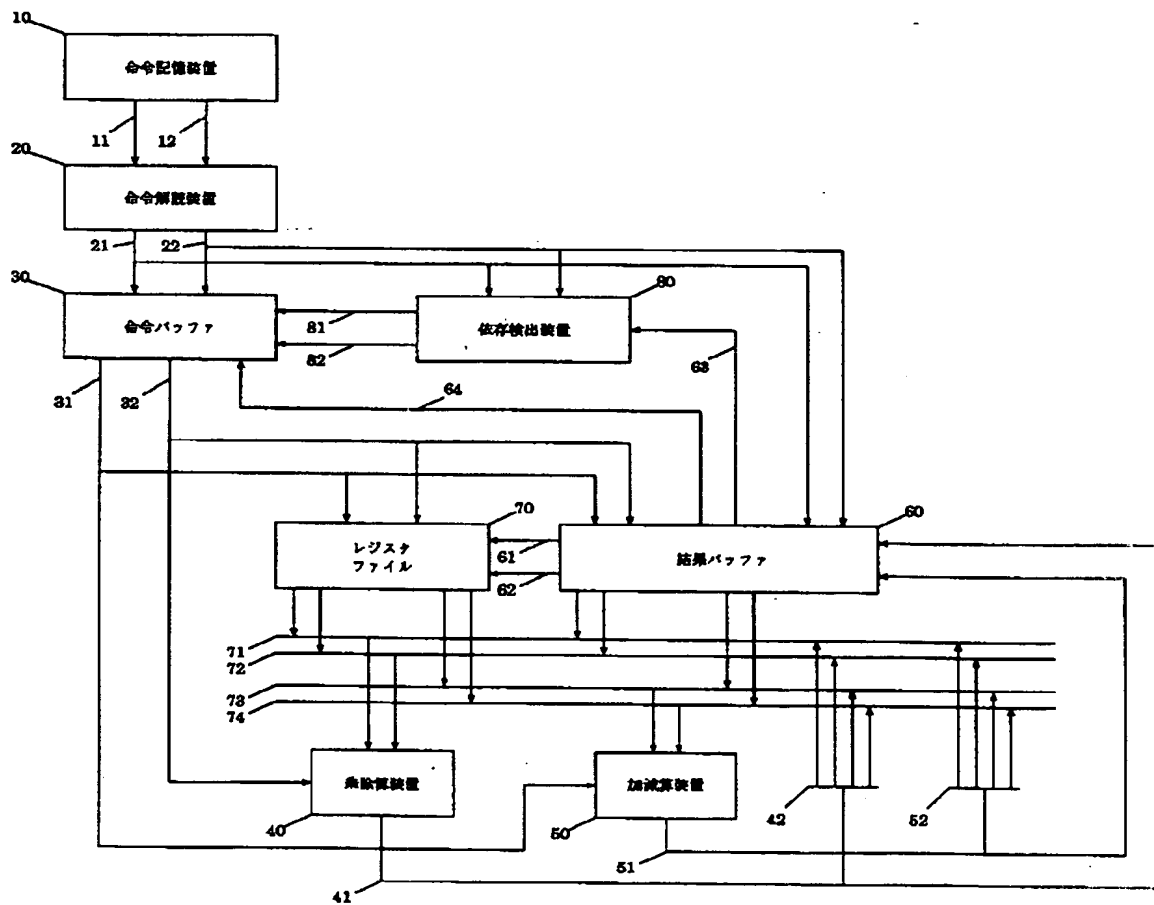
	1	2	3	4	5	6	7	8	9	10	11	12	13
7Fh 命令													
0x00 mul r1, r2, r3	D	L	E	E	S	R							
0x04 mul r3, r4, r5	D	B	B	L	E	E	S	R					
0x08 mul r11, r12, r13		D	B	B	B	L	E	E	S	R			
0x0c add r13, r14, r15		D	B	B	B	B	B	L	E	E	S	R	



【図3】

	valid	op	rs1	rs2	result_ptr	dep[2]	ok	mul	add
#0	0								
#1	1	mul	r3	r4	#1	0 0	1	1	0
#2	1	mul	r11	r12	#2	0 0	1	1	0
#3	1	add	r13	r14	#3	1 0	0	0	1

【図5】



**data dependency**

	PC	valid	rd	d0[2]	d1[2]	d2[2]	d3[2]	results	dv
#0	0x00	1	r3	0   0   0	0   0   0	0   0   0	0   0   0	(unknown)	0
#1	0x04	1	r5	0   0   0	0   0   0	0   0   0	0   0   0	(unknown)	0
#2	0x08	1	r13	0   0   0	0   0   0	0   0   0	0   0   0	(unknown)	0
#3	0x0c	1	r15	0   0   0	0   0   0	1   0   0	0   0   0	(unknown)	0
#4									
#5									
#6									
#7									